

U.S.P.S. EXPRESS MAIL LABEL NO.: EV 304938768 US

DATE OF DEPOSIT: August 7, 2003

Attorney Docket No. 15087US02

**ELECTRONIC DEVICE WITH AN UPDATE AGENT THAT EMPLOYS  
PREPROCESSING TECHNIQUES FOR UPDATE**

**RELATED APPLICATIONS**

[01] This application makes reference to, claims priority to and claims benefit from United States Provisional Patent Application Serial No. 60/405,253, entitled "Firmware Update Network And Process Employing Preprocessing Techniques", filed on August 22, 2002, United States Provisional Patent Application Serial No. 60/415,620, entitled "System For Generating Efficient And Compact Update Packages", filed on October 2, 2002, United States Provisional Patent Application Serial No. 60/441,867, entitled "Mobile Handset Update Package Generator That Employs Nodes Technique", filed on January 22, 2003, United States Provisional Patent Application Serial No. 60/447,977, entitled "Update Package Generator Employing Partial Predictive Mapping Techniques For Generating Update Packages For Mobile Handsets", filed on February 18, 2003, and United States Provisional Patent Application Serial No. 60/461,248, entitled "Electronic Device With An Update Agent That Employs Preprocessing Techniques For Update", filed on April 8, 2003.

**INCORPORATION BY REFERENCE**

[02] The complete subject matter of each of the above-referenced United States Patent Applications is hereby incorporated herein by reference, in its entirety. In addition, this application makes reference to United States Provisional Patent Application Serial No. 60/249,606, entitled "System and Method for Updating and Distributing Information", filed November 17, 2000, and International Patent Application Publication No. WO 02/41147 A1, entitled "System and Method for Updating and Distributing Information", publication date

May 23, 2002, the complete subject matter of which are hereby incorporated herein by reference, in their entirety.

#### FEDERALLY SPONSORED RESEARCH OR DEVELOPMENT

[03] [Not Applicable]

[MICROFICHE/COPYRIGHT REFERENCE]

[04] [Not Applicable]

#### BACKGROUND OF THE INVENTION

[05] Electronic devices such as, for example, mobile phones and personal digital assistants (PDA's) contain firmware and application software that are provided by the manufacturers of the electronic devices, by telecommunication carriers, or by third parties. These firmware and application software often contain software errors or "bugs". In addition, software developers may add new features over the life of the software. Therefore, new versions of the firmware and software are periodically released to fix the bugs, to introduce new features, or both.

[06] The process of updating such a device is relatively complex, and there are many potential sources of interruption of the update process. These include exhaustion of the battery operating the device, loss of the communication link used for the update, and interruption by outgoing or incoming calls. If the firmware/software of a mobile handset is updated and the mobile handset becomes inoperative, the user is likely to be disappointed with the service that disseminated the associated firmware/software update. For these reasons, firmware/software updates in mobile handsets need to be fault-tolerant. Unfortunately, these devices are constrained in many ways, and achieving fault-tolerant behavior is not easy, requiring great care in the design and management of related activities.

[07] Further limitations and disadvantages of conventional and traditional approaches will become apparent to one of ordinary skill in the art through comparison of such systems with

the present invention as set forth in the remainder of the present application with reference to the drawings.

## BRIEF SUMMARY OF THE INVENTION

[08] Aspects of the present invention may be found in a method of updating an electronic device from a first code version to a second code version according to a bank order, the electronic device having a non-volatile memory comprising a reserved area and a plurality of banks containing the first code version. Such a method may comprise moving the contents of the last bank in the bank order to the reserved area of the non-volatile memory. The method may also comprise shifting the contents of each bank in the bank order to the next bank in the bank order beginning with the penultimate bank in the bank order, and proceeding in descending bank order until the contents of the first bank in the bank order has been shifted to the second bank in the bank order. The method may comprise converting the contents of each bank in the bank order from the first code version to the second code version in a fault tolerant manner beginning with the second bank in the bank order and proceeding in ascending bank order until the last bank in the bank order has been converted, each converted bank being stored in the previous bank in the bank order.

[09] The method may further comprise transforming the contents of the reserved area from the first code version to the second code version in a fault tolerant manner, the second code version being stored in the bank that is last in the bank order. At least one of the moving and the shifting may be performed coincident with at least one preprocessing technique. The at least one preprocessing technique may use at least one preprocessing instruction, and may comprise at least one of a bubbles technique, a nodes technique, and a shift region technique. The updating may use only two writes to each non-volatile memory location being updated.

[10] An embodiment of the present invention may also comprise receiving an update package comprising at least one of at least a preprocessing instruction, an update instruction, and a bank order specification, where the receiving may use a public network and the receiving may use a wireless network. At least one of the converting and the transforming may use at least one update instruction, and at least one of the moving and shifting may use an offset of more than one bank.

[11] Additional aspects of the present invention may be found in a method of operating an electronic device, the electronic device having a non-volatile memory comprising a reserved area and a plurality of banks containing a first code version. An embodiment of the present invention may comprise moving the contents of a designated bank to the reserved area of the non-volatile memory, the designated bank thereby becoming an unoccupied bank. In addition, the embodiment may comprise shifting the contents of each of the plurality of banks other than the designated bank from an original bank to an unoccupied bank in a bank by bank fashion, each original bank thereby temporarily becoming an unoccupied bank. An embodiment of the present invention may further comprise converting the contents of each of the plurality of banks other than the designated bank from the first code version to a second code version in a fault tolerant manner beginning with the contents of the last bank shifted and proceeding in reverse order of the shifting. The second code version of each bank may be stored into the original bank from which the first code version of the bank was shifted. An embodiment in accordance with the present invention may also comprise transforming the contents of the reserved area of the non-volatile memory from the first code version to a second code version in a fault tolerant manner, the second code version being stored in the designated bank.

[12] In an embodiment of the present invention, at least one of the moving, shifting, converting, and transforming may be performed according to a specified bank order, and at least one of the moving and shifting may comprise preprocessing the contents of at least one of the plurality of banks. The preprocessing may comprise at least one of rearranging the contents of a bank, updating an address, updating a reference, and updating a branch instruction, and at least one of the converting and transforming may use at least one update instruction. The preprocessing may use at least one of a bubbles technique, a nodes technique, and a shift regions technique, and may use at least one preprocessing instruction.

[13] An embodiment of the present invention may further comprise receiving an update package comprising at least one of a preprocessing instruction, an update instruction, and a bank order specification, where the receiving may use a public network, and the receiving may use a wireless network.

[14] Further aspects of the present invention may be found in a method of updating an electronic device having a non-volatile memory comprising at least a first bank and a second bank, the at least a first bank and a second bank containing a first code version. An embodiment in accordance with the present invention may comprise moving the contents of the second bank to a reserve bank, transferring the contents of the first bank to the second bank, and converting the contents of the second bank to an updated version of the first bank. The method may further comprise storing the converted contents of the second bank into the first bank, transforming the contents of the reserve bank into an updated version of the second bank, and copying the transformed contents of the reserve bank to the second bank.

[15] At least one of the converting and the transforming may use at least one update instruction, and at least one of the moving and the transferring may comprise preprocessing the contents of at least one of the plurality of banks. The preprocessing may comprise at least one of rearranging the contents of a bank, updating an address, updating a reference, and updating a branch instruction. The preprocessing may use at least one of a bubbles technique, a nodes technique, and a shift regions technique, and may use at least one preprocessing instruction.

[16] At least one of the moving, transferring, converting, storing, transforming, and copying may use a specified bank order. In addition, the method may comprise receiving an update package comprising at least one of a preprocessing instruction, an update instruction, and a bank order specification, where the receiving may use a public network, and the receiving may use a wireless network.

[17] Yet other aspects of the present invention may be observed in a method of updating an electronic device from a first code version to a second code version, the electronic device having a non-volatile memory comprising a plurality of banks containing the first code version, the method comprising converting the first code version to the second code version in a fault tolerant manner, wherein the method requires only two writes to each bank being updated.

**[18]** These and other features and advantages of the present invention may be appreciated from a review of the following detailed description of the present invention, along with the accompanying figures in which like reference numerals refer to like parts throughout.

## BRIEF DESCRIPTION OF THE DRAWINGS

[19] Fig. 1 is a block diagram of an exemplary update network for dispensing an update package to an electronic device, the electronic device employing an update agent to update its operating system, applications software, and/or firmware, in accordance with the present invention.

[20] Fig. 2 is a block diagram depicting the operation of an exemplary “first bank bubble”-based preprocessing technique that may be employed by an update agent in an electronic device such as the electronic device illustrated in Fig. 1, in accordance with the present invention.

[21] Fig. 3 is a block diagram depicting the operation of another exemplary “first bank bubble”-based preprocessing technique that may be employed by an update agent in an electronic device such as the electronic device illustrated in Fig. 1, in accordance with the present invention.



## DETAILED DESCRIPTION OF THE INVENTION

[22] Aspects of the present invention relate to the updating of firmware/software components in electronic devices such as mobile handsets and personal digital assistants. More specifically, aspects of the present invention relate to a fault tolerant method of updating memory involving preprocessing the memory prior to performing an update in order to reduce the amount of data copied, the time needed to perform the update, and to extend the life of the updated memory.

[23] Fig. 1 is a block diagram of an exemplary update network 105 for dispensing an update package 123 to an electronic device 107, the electronic device 107 employing an update agent 113 to update its operating system 119, applications software 121, and/or firmware 117, in accordance with the present invention. The update package 123 is produced by a generator 131 and processed by the update agent 113, to update the firmware and software of electronic device 107 in a fault-tolerant mode employing bank-by-bank update techniques.

[24] The network 105 comprises an electronic device 107, a device server 127, an update store 129, and a generator 131. The electronic device 107 is communicatively coupled to the device server 127, which is communicatively coupled to the update store 129. The communication link between the electronic device 107 and the device server 127 may be, for example, a wired network, a wireless network, a public network, an Internet-based network, and an intranet-based network, or any combination of the above. The update store 129 has stored within it a collection of update packages produced by the generator 131. The update packages stored within update store 129 are communicated to device server 127 for distribution to and processing by the electronic device 107. In one embodiment of the present invention, the device server 127, the update store 129, and the generator 131 may be co-located on a single computer or grouped in any combination on a networked cluster of computers.

[25] The electronic device 107 comprises a volatile random access memory (RAM) 125 and a non-volatile memory 109. The volatile memory 125 may be used for the execution of

programs and for the temporary storage of program variables. The non-volatile memory 109 is comprised of a plurality of banks or blocks and may be used for the storage of programs, update packages, and for other non-volatile storage. The non-volatile memory 109 comprises a loader 111, an update agent 113, a preprocessor 115, a firmware 117, an operating system (OS) 119, one or more applications 121, and at least one update package 123. In a particular embodiment of the present invention the preprocessor 115 may also be a part of the update agent 113.

[26] In one embodiment of the present invention, the generator 131 may process a first version and a second version of the firmware of the electronic device 107 in order to produce a set of preprocessing instructions, and difference information comprising a set of update instructions. The preprocessing instructions and the update instructions may be assembled into an update package. The update packages generated by the generator 131 may be stored in the update store 129 for subsequent retrieval by the electronic device 107, via the device server 127. Once retrieved from the update store 129, an update package may be used by electronic device 107 to convert or transform portions of its firmware/software from the first version to the second version in a bank-by-bank fashion.

[27] The update agent 113 in an embodiment of the present invention may employ the preprocessor 115 as a part of the processing of the update package 123. In such an embodiment, the preprocessor may shift selected banks of non-volatile memory 109 by a distance equal to, for example, the size of one or more banks, according to a bank order. In this discussion, the terms “bank” or “block” may be used to represent a portion of non-volatile memory 109. The shifting of selected banks of the non-volatile memory 109 allows an embodiment of the present invention to eliminate the need to “back up” the bank being updated during a subsequent fault tolerant update process. An example fault tolerant update process is described in U.S. Provisional Patent Application Serial No. 60/434,712 (attorney docket number 14323US02), entitled “Mobile Handset With A Fault Tolerant Update Agent”, filed December 18, 2002, the complete subject matter of which is hereby incorporated by reference herein, in its entirety.

[28] For example, prior to performing a fault tolerant update, the preprocessor 115 may move the firmware 117 and/or software 119, 121 portions of the non-volatile memory 109 by the size of at least one bank, initially creating an unoccupied first bank or “first bank bubble”. This may be done as a part of other preprocessing operations such as, for example, a “bubbles” preprocessing technique, a “nodes” preprocessing technique, or a “shift regions” preprocessing technique, and may be performed according to a bank order specification contained within an update package. An example of a suitable “bubbles” preprocessing technique may be found in United States Provisional Patent Application Serial No. 60/405,253, entitled “Firmware Update Network And Process Employing Preprocessing Techniques”, filed on August 22, 2002. An example of another “bubbles”-type preprocessing technique may be found in United States Provisional Patent Application Serial No. 60/415,620, entitled “System For Generating Efficient And Compact Update Packages”, filed on October 2, 2002. An example of a “nodes” preprocessing technique may be found in United States Provisional Patent Application Serial No. 60/441,867, entitled “Mobile Handset Update Package Generator That Employs Nodes Technique”, filed on January 22, 2003. Finally, a suitable “shift regions” preprocessing technique is described in United States Provisional Patent Application Serial No. 60/447,977, entitled “Update Package Generator Employing Partial Predictive Mapping Techniques For Generating Update Packages For Mobile Handsets”, filed on February 18, 2003. The complete subject matter of each of the United States Patent Applications listed above is hereby incorporated herein, in its entirety. The unoccupied bank or “bubble” may then be used by the update agent 113 as a moving bubble or destination bank into which the updated firmware/software banks are written, after each of them have been updated in the RAM 125. An embodiment of the present invention may combine the shifting of a set of banks of firmware with other preprocessing activities such as, for example, the updating of addresses, references, link instructions, etc. that may be performed during the preprocessing methods of the above references. The “first bank bubble” technique described herein may be used whether or not other preprocessing techniques are employed.

[29] Fig. 2 is a block diagram depicting the operation of an exemplary preprocessing technique that may be employed by an update agent in an electronic device such as the electronic device 107 illustrated in Fig. 1, in accordance with the present invention. The illustration of Fig. 2 shows a non-volatile memory 209 having banks 1 to N, prior to the application of the “first bank bubble” preprocessing technique, and a non-volatile memory 211 having banks 1 to N and bubble 213 corresponding to the non-volatile memory 209, following the application of the “first bank bubble” preprocessing technique. In the exemplary embodiment illustrated in Fig. 2, a bank order specification is not used. A preprocessor such as the preprocessor 115 of Fig. 1 may shift selected banks of the firmware contained in non-volatile memory 209 by the size of one or more banks. This shift of the contents of the non-volatile memory 209 creates an empty bank or “bubble”, illustrated in Fig. 2 as bubble 213 in the non-volatile memory 211. The shifting of the banks of non-volatile memory 209 may be performed by an update agent, such as the update agent 113 of Fig. 1, as a part of performing other preprocessing operations described above. The illustration of Fig. 2 shows a situation where all N banks of the non-volatile memory 209 are modified in a sequential fashion beginning with bank 1. An embodiment of the present invention may also be applied to updates of memory, such as non-volatile memory 209, where not all banks within the memory are being updated, and where the order of the bank update is not a sequential order.

[30] As described above, an explicit bank order specification may not be included in the update package. In such an embodiment, the update agent may first move one or more banks of non-volatile memory 209 in a sequential fashion, as shown in Fig. 2. The movement of these banks may be combined with other preprocessing operations and may begin by moving the contents of bank N to a designated or reserved area of non-volatile memory 209. Bank N-1 may then be moved to bank N, bank N-2 moved to bank N-1, and so on until bank 1 is moved to bank 2. In this manner, an empty/unoccupied bank or “bubble” is created at bank 1.

[31] Following the shifting of the banks to be updated, the update agent may begin updating the non-volatile memory starting with bank 1. The update agent may copy the

contents of bank 2 to a designated “working bank” in volatile memory, such as the RAM 125 of Fig. 1. The update agent may then use update instructions for bank 1 that are contained in the update package to update the contents of the working bank. Upon updating the working bank, the update agent may store the contents of the working bank into the currently unoccupied bank 1. The update agent may use a cyclic redundancy check (CRC), MD5 checksum, or other calculated value to verify the success of the update. The update agent may then update the now unoccupied bank 2, using the original contents of bank 2 that was stored in bank 3, and continue in like manner until bank N-1 is updated. Finally, bank N may be updated using the original bank N contents that was stored in the designated reserved or backup location. In this manner, an embodiment of the present invention may combine the shifting of a set of banks of firmware with other preprocessing activities as described above, reducing the number of writes to each updated memory location from four writes per updated memory location to two writes per updated memory location, while also providing complete fault tolerance. This minimizes the number of writes to non-volatile memory, reducing the time needed to update the memory, and extending the life of the non-volatile memory devices that are typically used in such applications.

[32] Fig. 3 is a block diagram depicting the operation of another exemplary “first bank bubble”-based preprocessing technique that may be employed by an update agent in an electronic device such as the electronic device 107 illustrated in Fig. 1, in accordance with the present invention. The illustration of Fig. 3 shows a first, original arrangement of a non-volatile memory 309, and a second, preprocessed arrangement of the non-volatile memory 311. In the example illustrated in Fig. 3, the update package received by the electronic device 107 of Fig. 1 includes a bank order specification (a.k.a., “bank order”). The bank order may be defined by a generator such as generator 131 of Fig. 1 at the time the update package is created. The use of a bank order specification in the update process may permit a generator, such as the generator 131 of Fig. 1, to produce a more compact and efficient update package. A preprocessor, such as the preprocessor 115 of Fig. 1, may use the bank order contained in an update package to move one or more banks of a non-volatile memory such as non-volatile memory 309 prior to performing the update of the affected banks. In an

embodiment of the present invention, a preprocessor such as the preprocessor 115 of Fig. 1 may also be contained within an update agent, such as the update agent 113 of Fig. 1. As described above, the shifting operations used by an embodiment of the present invention may be in addition to and combined with other operations indicated in the update package.

[33] Using the bank order specification, the preprocessor may first move the contents of the last bank in the bank order to a designated reserved or backup bank in the arrangement of non-volatile memory 311. The preprocessor may then move the bank in position N-1 in the bank order to the location of the bank in position N in the bank order, then move the bank in position N-2 in the bank order to the location of the bank in position N-1 in the bank order, and so on. This sequence of moves continues until the bank that is first in the bank order is moved to the memory location of the bank that is second in the bank order. In this manner, an empty or unoccupied bank or “bubble” is created that corresponds to the first bank in the bank order.

[34] For example, let us assume that a bank order specification is contained in the update package to be used in updating the non-volatile memory 309 of Fig. 3. Let us further assume that the bank order specification is as shown in Fig. 3, (2, N-1, 3, 1, N), indicating that the banks of non-volatile memory 309 are to be updated in the order bank 2, bank N-1, bank 3, bank 1, and bank N. In an embodiment of the present invention, a preprocessor, such as the preprocessor 115 of Fig. 1, may use the bank order specification to move the affected banks of non-volatile memory 309 in the reverse of the bank order specification. As shown in the arrangement of non-volatile memory 311 of Fig. 3, the last bank in the bank order specification, bank N of the arrangement of non-volatile memory 309, is moved to a reserved or “backup” bank in the preprocessed arrangement of the non-volatile memory 311. Then, the next-to-last bank in the bank order specification, bank 1 of the arrangement of non-volatile memory 309, is moved to the vacant bank created when bank N was moved to the backup bank in the arrangement of non-volatile memory 311. Next bank 3 in the arrangement of non-volatile memory 309 is moved to the vacant bank created when bank 1 was move to bank N of the arrangement of nonvolatile memory 311. This sequence of bank-to-bank moves continues until the first bank in the bank order specification, in this case bank

2, is moved to reside in the original location of the second bank in the bank order, in this case bank N-1. The result is a vacant/unoccupied bank or “bubble” 313 at the original location of the first bank in the bank order, bank 2.

[35] Following the shifting of the banks to be updated, an update agent, such as the update agent 113 of Fig. 1, updates the preprocessed non-volatile memory 311 starting with the first bank in the bank order. The update agent may first copy the contents of the second bank in the bank order, bank N-1, that contains the contents of the first bank in the bank order, bank 2, to a designated “working bank” in volatile memory, such as the RAM 125 of Fig. 1. The update agent 113 may then use the update instructions contained in the update package that apply to the first bank in the bank order, in this case bank 2, to update the contents of the working bank. In one embodiment, the update may be performed by converting or transforming the contents of the working bank using update instructions contained in an update package. Upon updating the working bank, the update agent may copy the contents of the working bank into the first (currently unoccupied) bank in the bank order, bank 2. The update agent may calculate a cyclic redundancy check (CRC), MD5 checksum, or other value to verify the success of each of the steps of the update. The update agent may then update the now unoccupied second bank in the bank order in the same manner, using the original contents of the second bank that is stored in the third bank in the bank order. The update continues in like manner until the next-to-last bank in the bank order is updated. Finally, the last bank in the bank order is updated, in this example bank N, using the original bank N contents stored in the designated reserved or backup location. In this manner, an embodiment of the present invention reduces the amount of copying of the contents of the memory being updated, reducing the time needed to update the memory, and extending the life of the memory devices that are typically used in such applications.

[36] As discussed above, the preprocessing activities defined by the generator 131 of Fig. 1 and described in the update package received by the electronic device 107 of Fig. 1 may be combined with the shifting operations described above with respect to Figs. 2 and 3. In the process described above, banks of a non-volatile memory 309 are moved by preprocessing activities to create an empty or unoccupied bank, shown as bubble 313 in the arrangement of

non-volatile memory 311. In the discussion above and the associated illustrations of Figs. 2 and 3, this shifting creates an unoccupied bank or “bubble” equal to the size of one bank. In another embodiment, however, the shifting of firmware/software code using other bubble sizes, such as the size of multiple banks, is also possible.

[37] Aspects of the present invention provide a network having a plurality of electronic devices and that employs a generator to produce update packages for the electronic devices. A corresponding update agent in the electronic devices uses the update packages to update its firmware and/or software. The generator specifies preprocessing steps that incorporate shifting of firmware and/or software code by one or more memory banks so as to create a “first bank bubble” in the non-volatile memory of the electronic device during a preprocessing activity. A preprocessor or update agent in the electronic device subsequently employs the “first bank bubble” as a moving bubble or vacant bank to conduct an efficient and fault tolerant update.

[38] While the present invention has been described with reference to certain embodiments, it will be understood by those skilled in the art that various changes may be made and equivalents may be substituted without departing from the scope of the present invention. In addition, many modifications may be made to adapt a particular situation or material to the teachings of the present invention without departing from its scope. Therefore, it is intended that the present invention not be limited to the particular embodiment disclosed, but that the present invention will include all embodiments falling within the scope of the appended claims.